

(NASA-CR-139561) FINITE PURE INTEGER
PROGRAMMING ALGORITHMS EMPLOYING ONLY
HYPERSPHERICALLY DEDUCED CUTS Final
Report (Rice Univ.) 26 p HC \$~~5.75~~ 3.75

N74-30550

CSCL 09B G3/08 46188

Unclas

Final

FINITE PURE INTEGER PROGRAMMING
ALGORITHMS EMPLOYING ONLY HYPER-
SPHERICALLY DEDUCED CUTS



Richard D. Young
Departments of Economics and
Mathematical Sciences
Rice University
Houston, Texas 77001
May 3, 1971

This is a working paper and should not be quoted without permission.

This research has been wholly supported by NASA Grant NGR-44-006-128

ABSTRACT

This paper develops three algorithms that may be based exclusively on hyperspherically deduced cuts. The algorithms only apply, therefore, to problems structured so that these cuts are valid. The algorithms are shown to be finite.

The general strategy of the methods is simple. Let S denote the solution set for the associated linear program and let θ denote the hypersphere that contains S , while $\bar{\theta}$ represents the boundary of θ . T_i symbolizes the intersection of all the half space restrictions that are adjoined through cycle $i - 1$. The current (reduced) solution set is S_i . In every cycle $T_i \supset S_i$ and for some finite i (unless an integer solution is already located) we obtain $T_i \subset \theta$ and $T_i \cap \bar{\theta} = \emptyset$ which implies T_i (and therefore S_i) contains no integer solution.

1. Introduction and Summary.

This paper describes two primary algorithms and one composite algorithm for a special class of pure integer programs. The special class of problems consists of problems that admit hyperspherically deduced cuts.

Let the integer program be given as

$$\begin{aligned} &\max c^t x \\ &\text{subject to: } x \in S \\ &x \text{ to be an all integer vector.} \end{aligned}$$

S is a convex polyhedron in E^n and c^t is an integer vector. For our special class of problems, we have

$$S \subset \theta$$

where θ is a closed hypersphere, and

$$x^1 \text{ is an integer solution in } S \Leftrightarrow x^1 \in S \cap \bar{\theta}$$

where $\bar{\theta}$ is the boundary of θ .

Problems in this class permit use of special cuts. The theory behind these cuts, generalizations of the hypersphere to more general convex sets containing S , and discussion of the scope of the special class of problems is found in references [1], [2], [3] and [5].

The algorithms developed here are shown to be finite and do not depend on the use of Gomory cuts to insure finiteness. Unfortunately, the finiteness of these methods is also substantially independent of special characteristics of the hyperspherically deduced cuts. Finiteness will be achieved even though cuts considerably weaker than the hyperspherically deduced cuts are used. This result is achieved at the cost of employing tree-search methods

in certain steps of the algorithms, with all the implicit attendant difficulties of data storage and search.

These algorithms are not designed to solve the integer programming problem directly. Rather, the goal is more elementary: to locate a single integral $x^1 \in S$ or determine that $S \cap \bar{\theta} = \emptyset$ and S has no integer solutions.

It is completely straightforward to use these algorithms to construct methods for solving $\max c^t x$ subject to $x \in S$, and x an integer point. For example: if S contains no integer point x^1 , then the problem has no solution. If an integral $x^0 \in S$ is located by one of our algorithms, then we repeat the procedure by attempting to locate an integral $x \in S_1 \equiv (S \cap \{x | c^t x \geq c^t x^0 + 1\})$. Repetitive application of this routine will obviously solve the original integer program.

The succeeding sections discuss first the general characteristics of the two primary algorithms, and then the specifics of algorithms I and II. An exemplary composite method, algorithm III, is developed next. Following that, the finiteness argument is stated. The final sections discuss related work and future research possibilities related to the developments of this paper.

2. Notational Conventions.

This paper will employ the following notational conventions:

- Vectors will be considered to be column vectors and the superscript t will always denote transposition.
- The sets S and θ will be assumed to be subsets of Euclidean n -space, denoted by E^n .
- The origin will be assumed to be located at the center of θ .
- The radius of θ is $(p)^{\frac{1}{2}}$. Thus for any point $y \in \bar{\theta}$ we have $y^t y = p$.
- The symbols x , y , z , and h will all be used for vectors in E^n . Typically x will denote an element of S , and y an element of θ while z is neutral with respect to these classifications - a utility infielder. Normally h will denote the fixed coefficients of (or a normal to) a hyperplane equation.
- Since the constraints of interest here will always non-trivially intersect θ , if $h \in \bar{\theta}$ then the constraint $h^t y \leq v$ will have $-p < v < p$.

3. A General Outline of the Algorithms.

Both of the primary algorithms (I and II) developed in this paper have common procedural regularities that are outlined in this section. Subsequent sections will introduce the procedural details that distinguish the two particular methods. The typical cycle of these algorithms consists of two phases. I shall label these phases the S-cut phase and the θ -tangent phase.

3a. S-Cut Phase.

In the S-cut phase an extreme point x^1 , of S_1 (the current, reduced remnant of S) is located. If x^1 is an integer point the procedure terminates. Otherwise we generate a cut that removes x^1 from S_1 . Denote by H_1 the feasible halfspace of this cut. H_1 is then used to reduce S_1 to S_{1+1} (i.e., $S_{1+1} = S_1 \cap H_1$), and to effect a similar reduction for the θ -tangent phase.

Location of x^1 in the S-cut phase is accomplished by solving a linear program. The solution set of this linear program is S_1 . The objective function may be selected arbitrarily in the first cycle and thereafter this datum is provided as an output from the θ -tangent phase.

3b. θ -Tangent Phase.

The θ -tangent phase serves two purposes: (i) to provide a measure of the possibility that integer points exist in S_1 (hence in S) and (ii) to locate prospectively reasonable "regions" of S_1 in which integer solutions may be sought.

Let $T_1 = \bigcap_{k=1}^{i-1} H_k$. Clearly $S_1 \subset T_1$.

The goal of this phase is to find a point $y^1 \in (T_1 \cap \bar{\theta})$. If $T_1 \cap \bar{\theta}$

is empty then $S_1 \cap \bar{\theta}$ is empty and no integer solutions exist in S_1 or in S . If a point $y^1 \in (T_1 \cap \bar{\theta})$ is located, then y^1 is used to generate an objective function for the next iteration of the S-cut phase.

To see the rationale for this use of y^1 , note that $y^1 \in T_1$ implies y^1 has not yet been eliminated from S (if $y^1 \in S$) and $y^1 \in \bar{\theta}$ implies y^1 is an integer solution (if $y^1 \in S$). Hence y^1 is a prospective integer solution. Now let $h^1 = y^1$ denote the vector of coefficients for the supporting hyperplane¹ (of θ) at y^1 . If y^1 is in S then y^1 will be the optimal solution to

$$\max h^1 \cdot x \quad \text{subject to } x \in S_1,$$

which is the problem (induced by h^1) in the S-cut phase.

¹ $h^1 = y^1$ follows from the center of θ being the origin. y^1 is a vector from the origin to a point in $\bar{\theta}$. The set $\{y | (y^1)^t y = p\}$ is the tangent hyperplane to θ at y^1 .

4. Specific Description of Algorithm I.

The description of algorithm I is given first in general summary form. Following that, specific methods of implementing certain steps in the algorithm are discussed.

4a. Procedural Outline.

Initial Step

$$T_0 = E^n$$

$$S_0 = S$$

Repeated Step

θ -Tangent Phase; index = i

1. If $T_i \cap \bar{\theta} = \emptyset$ stop: S contains no integer solutions.

2. If $T_i \cap \bar{\theta} \neq \emptyset$, locate $y^i \in (T_i \cap \bar{\theta})$.

(Steps 1 and 2 are accomplished by solution and analysis of an auxiliary problem described below).

3. Define $h^i = y^i$.

4. Go to the S-cut phase.

S-Cut Phase; index = i

1. Solve: $\max(h^i)^t \cdot x$ subject to $x \in S_i$.

Denote the optimal solution by x^i .

2. If x^i is an integer solution, stop. A solution has been obtained.

3. If x^i is not an integer solution, generate a new halfspace restriction

$$H_i = \{z \mid (h^i)^t z \leq v_i\}, \text{ where } v_i \text{ is a scalar}$$

satisfying $\bar{v}_1 > v_1 \geq \underline{v}_1$.

(Definition of \bar{v}_1 and \underline{v}_1 is given below.)

4. Redefine $S_{i+1} = S_i \cap H_i$

$$T_{i+1} = T_i \cap H_i$$

Advance the index to $i + 1$, and go to θ -tangent phase.

4b. Discussion

There are two main ambiguities in this statement of algorithm I. Step 2 of the θ -tangent phase requires "location" of $y^i \in (T_i \cap \bar{\theta})$. How is this step accomplished? When $i = 0$ and $T_0 = E^n$, an arbitrary point on $\bar{\theta}$ may be selected. When $i > 0$, solution and analysis of the auxiliary problem described below in section 4c provides one method for locating y^i or determining that $T_i \cap \bar{\theta}$ is empty.

The second ambiguity involves the depth of the cut $-(h^i)^t x \leq v_i$ - adjoined in step 3 of the S-cut phase. The maximum depth of the cut can be determined by considering the points where edges incident at x^i intersect $\bar{\theta}$. Let K denote an index set for the edges incident at x^i ; and let x^{ik} symbolize the point of intersection of the k^{th} edge with $\bar{\theta}$. Define

$$\underline{v}_i = \max_{k \in K} (h^i)^t x^{ik}.$$

The theory justifying convexity cuts - see [1], [2], [5] - shows that if $v_i \geq \underline{v}_i$ the cut will remove no integer solutions from S_i . To remove some points from S_i the cut must satisfy

$$(h^i)^t \cdot x < (h^i)^t x^i = \bar{v}_i.$$

The range $[\underline{v}_i, \bar{v}_i)$ is permitted for v_i because setting $v_i = \underline{v}_i$ may

not be universally advantageous. Consider, for example, the relation between T_i and $T_{i+1} = (T_i \cap H_i)$. In the θ -tangent phase (in cycle $i + 1$) it is necessary to determine whether the new extreme points in T_{i+1} (and not in T_i) are in θ . This search may be easier to conduct when the cut that determines H_i is shallower.

4c. The Auxiliary Problem.

There are many ways to implement steps 1 and 2 of the θ -tangent phase, and while only one specific method is presented here, the possibility of modifying the algorithm by altering the implementation of these steps should be recognized.

We propose the solution of a problem that seeks a point in θ that, roughly speaking, maximizes the "distance" from the "closest" hyperplane boundary of T_i . Here "distance" from a hyperplane boundary is measured in terms of the magnitude of the slack variable associated with the halfspace constraint. If this maximin distance is negative then $T_i = \emptyset$ and hence $T_i \cap \bar{\theta}$ is empty. If the distance is positive and the maximal point is in $\bar{\theta}$, then we have found y^i . If the distance is positive, and the maximizing point is interior to θ , then a path of edges (hopefully short) will lead from the maximizing point to each corner point in T_i . These corner points (hopefully not too numerous) can be located and tested for membership in θ . This testing either finds a suitable $y^i \in \bar{\theta}$ or determines that $T_i \cap \bar{\theta} = \emptyset$.

Formally we solve the following auxiliary problem (PA):

$$\max u$$

$$\text{subject to } (h^k)^t \cdot y + r_k + u = v_k, \quad (k = 0, 1, \dots, i-1)$$

$$y^t \cdot y \leq p,$$

$$y \text{ unrestricted}$$

$$u \text{ unrestricted}$$

$$r_k \geq 0 \quad (k = 0, 1, \dots, i-1).$$

Let the optimal solution to (PA) be denoted by $\bar{y}, \bar{u}, \bar{r}$, (where r symbolizes the vector of slacks from (PA)). Consider the following three cases:

- I. $\bar{u} < 0$,
- II. $\bar{u} \geq 0$ and $\bar{y}^t \cdot \bar{y} = p$,
- III. $\bar{u} \geq 0$ and $\bar{y}^t \cdot \bar{y} < p$.

These cases have the following interpretations:

- I. implies $T_i \cap \bar{\theta}$ is empty,
- II. implies $\bar{y} \in (T_i \cap \bar{\theta})$. Designate $y^t = \bar{y}$,
- III. requires a subsidiary analysis of the solution set for (PA).

In case III we ignore the nonlinear constraint and investigate basic feasible solutions to (PA) that have $u = 0$. These solutions are the corner points of T_i . Let $\hat{y}, \hat{u}, \hat{r}$ designate such a solution.

If any corner point solution has $\hat{y}^t \cdot \hat{y} \geq p$, then the set $T_i \cap \bar{\theta}$ is not empty. To see this note that both \hat{y} and \bar{y} are in T_i and \bar{y} is in the interior of θ while \hat{y} is not interior to θ . Accordingly there exists a point \tilde{y} , a convex combination of \bar{y} and \hat{y} , such that $\tilde{y} \in \bar{\theta}$ and $\tilde{y} \in T_i$. Designate $y^t = \tilde{y}$.

Alternatively every corner point, $\hat{y}, \hat{u}, \hat{r}$, may have $\hat{y}^t \cdot \hat{y} < p$. This means that all the corner points of T_i are in the interior of θ . Since both T_i and θ are convex we have $T_i \cap \bar{\theta} = \emptyset$.

While it is not my purpose to discuss in this paper detailed methods of conducting the search alluded to in the above discussion of Case III, I believe a brief comment is fitting on the fact that some cycles of this algorithm require an exhaustive investigation of all basic solutions to (PA)

that have $u = 0$ (ignoring the nonlinear constraint). Computationally this is certainly repugnant. Still, some mitigating factors exist and should be mentioned. These are:

- i) An integer solution may be located before Case III is encountered.
- ii) If the space containing T_1 and θ has n dimensions, then at least $n + 1$ cuts must be present in (PA) before Case III can occur.
- iii) In the first occurrence of Case III there are only $n + 1$ extreme points with $u = 0$. If in this instance $T_1 \cap \bar{\theta} \neq \emptyset$ and a cut must be added, it is only necessary, in the next cycle, to locate (anew) the newly created extreme points for which $u = 0$ and $r_1 = 0$. While this latter task may require extensive search, it is certainly possible and perhaps likely that only n new extreme points will be created by the new cut and that location of these new points will be comparatively straightforward.

Paradoxically it appears that locating all the new extreme points is entirely straightforward if the cut (that creates the new extreme points) is sufficiently shallow.

5. General Description of Algorithm II.

Algorithm I might be characterized as "omnidirectional" in the sense that the points of y^i generated in the θ -tangent phase tend to distribute somewhat uniformly over the surface of $\bar{\theta}$. In any case there is no built-in bias in the algorithm favoring search in any particular sector of $\bar{\theta}$.

Algorithm II, in contrast, has a definite directional orientation. The algorithm is specified in part, by an initial datum, the vector d . The search conducted by algorithm II examines potential solutions y^i in a sequence that has monotonically decreasing values for the expression $d^t y^i = u_i$.

This algorithm has the same general two phase cycle routine as algorithm I. The S-cut phase is identical in many respects: a tangent hyperplane to θ , denoted by h^i is an input from the θ -search phase. An extreme point of S_i is located by optimizing $(h^i)^t x$ over all $x \in S_i$. If the optimal point, x^i , is integral (in $\bar{\theta}$) the procedure terminates, otherwise the cut $(h^i)^t x \leq v_i$ is developed, and used to redefine S_{i+1} and T_{i+1} . The main difference in this phase is an additional updating step that is incidental to new developments in the θ -tangent phase.

The special characteristics of this algorithm reside mostly in the θ -tangent phase. In very general terms the procedure of this phase is as follows:

1. Locate the point in $T_i \cap \bar{\theta}$ that maximizes a fixed linear function $d^t y = u$. Let \bar{y}^* denote the maximizing $y \in (T_i \cap \bar{\theta})$.

2. Define $\underline{w} = \min_{x \in S} d^t x$.

If $d^t \bar{y}^* < \underline{w}$, stop: no integer solutions exist in S .

Otherwise designate $y^i = \bar{y}^*$.

The principal problem here is the search required in step 1. Locating the point that maximizes $d^t y$ subject to $y \in T_1 \cap \bar{\theta}$ is not straightforward -- because of the non-convexity of the set $T_1 \cap \bar{\theta}$. This difficulty is met, in part at least, by controlling the development of T_1 so that limited incremental enumeration will reveal the solution sought in step 1.

5a. Detailed Description of the Algorithm II.

Initial Step.

1. Develop n cuts $(h^i)^t y \leq v_i$ ($i = 1, \dots, n$), such that the point y^+ satisfying $(h^i)^t y^+ = v_i$, ($i = 1, \dots, n$) is interior to θ . (Such a start is always possible: e.g., choose a non-degenerate extreme point of S , and let the hyperplanes incident at this point be defined as $(h^i)^t y = v_i$ ($i = 1, \dots, n$)). We assume that the relation of the vector d^t to these n cuts is such that $d^t y^+ \geq d^t y$ for all y satisfying $(h^i)^t \cdot y \leq v_i$, ($i = 1, \dots, n$).
2. Define $S_1 = S \cap \{x | h^i x \leq v_i, i = 1, \dots, n\}$, and

$$T_1 = \{y | h^i y \leq v_i, i = 1, \dots, n\}.$$
3. List all edges incident at y^+ . Denote these edges by E_1, \dots, E_n . List also the points $\bar{y}^1, \dots, \bar{y}^n$ where these edges, considered as edges of T_1 , (feasibly) intersect $\bar{\theta}$, and finally, list the d -function values $d^t \bar{y}^i = u^i$, ($i = 1, \dots, n$).

Repeated Step.

θ -Tangent Phase; index = i .

1. Select from the list the edge E_* for which $d^t \bar{y}^* = u^* \geq u^j$ for all edges E_j on the list. If $u^* \leq \underline{w}$ stop: no integer points exist in S ; otherwise go to 2.

2. Designate $y^i = \bar{y}^*$. Define $h^i = y^i$.
3. Remove E_* , \bar{y}^* , and u^* from the list.
4. Go to the S-cut phase.

S-Cut Phase; index = i.

1. Solve $\max (h^i)^t x$, subject to $x \in S_i$. Denote the optimal solution to this problem by x^i .
2. If x^i is an integer solution, stop; otherwise generate a new restriction:

$$(h^i)^t x \leq v_i$$

where v_i is a scalar constant satisfying

$$\underline{v}_i \leq v_i < \bar{v}_i. \quad (\text{See definition of } \underline{v}_i \text{ and } \bar{v}_i \text{ above p. 7}).$$

3. Redefine $S_{i+1} = S_i \cap H_i$ and $T_{i+1} = T_i \cap H_i$ where $H_i \equiv \{z \mid (h^i)^t z \leq v_i\}$.
4. List the new edges, E , formed by the intersection of the boundary of H_i with planes of T_i . Also list the associated points \bar{y} where these new edges (feasibly) intersect $\bar{\theta}$, and the associated d-function values $d^t \bar{y} = u$.
5. Remove from the list any edges E (as well as \bar{y} and u) for which $\bar{y} \notin H_i$.
6. Go to the θ -tangent phase with index = $i + 1$.

5b. Discussion.

In the 4th and 5th steps of the S-cut phase we require the listing of all new edges of T_{i+1} that feasibly penetrate $\bar{\theta}$, as well as the discard of existing listed edges that no longer feasibly penetrate $\bar{\theta}$. No

specific method for solving this problem is offered here. It is clear that all new edges are in the boundary of H_1 and that all eliminated edges must penetrate H_1 . It is equally clear that a finite tree search will locate all new edges, and that a finite sequence of tests will detect all eliminated edges. Relatively efficient and adaptive methods undoubtedly exist for conducting these searches and for efficiently storing essential information required for the list, and by the list revision procedures. The omission of these details in this paper reflects (i) time pressure and (ii) the failure of several comparatively simple trial methods to be generally applicable. I hope to repair this omission in a later paper.

It should also be noted that the comprehensive listing of all edges that feasibly penetrate $\bar{\theta}$ is not essential to the method. It is only necessary to find the edge E that maximizes $d^t y$. This can be accomplished by less storage of edges and more search. The routine presented in the formal statement of the method was chosen primarily because it seemed amenable to more succinct expression.

It is worthwhile to emphasize that Algorithm II progresses by implicitly generating a sequence of legitimate cuts of the form $d^t y^i \leq u_i$ with u_i decreasing monotonically with respect to increases in i .

To see this, note first that initially T_1 is a cone, with its vertex interior to $\bar{\theta}$. The vertex, y^+ , maximizes $d^t y$ over all $y \in T_1$.

Consider the continuum of hyperplanes $d^t y = u$ where $u \leq d^t y^+$, and the continuum of intersections of these hyperplanes with T_1 . Denote the set $T_1 \cap \{y | d^t y = u\}$ by $D(u)$, where $u \leq d^t y^+$. Each set $D(u)$ is the convex hull of points on the edges that emanate from y^+ . The largest value of u for which $D(u)$ intersects $\bar{\theta}$ must be equal to the value $d^t y^*$, determined in step 1 of the $\bar{\theta}$ -tangent phase. Clearly y^* maximizes

$d^t y$ subject to $y \in T_1 \cap \bar{\theta}$, since for all $u > d^t \bar{y}^*$, $D(u) \cap \bar{\theta} = \emptyset$. This means that the implicit cut $d^t x \leq d^t \bar{y}^*$ is legitimate.

In subsequent cycles, the same essential relations portrayed above continue to hold. The cut written in the first iteration of the S-cut phase causes $D(d^t \bar{y}^*)$ to be reduced by intersection with H_1 , so that typically, u can be reduced below the value $d^t \bar{y}^*$ while $D(u)$ remains contained in the strict interior of θ . All new edges - created by intersection of T_1 with H_1 - that feasibly intersect $\bar{\theta}$ are listed. Hence it is possible to keep account of $D(u)$, which is the convex hull of points on the listed edges that have $d^t y = u$, and u can be decreased until the set $D(u)$ again intersects $\bar{\theta}$. The cycle is repeated until either an integer point is located in the S-cut phase, or until the value $d^t \bar{y}^*$ at some stage is less than \underline{w} .

It should be pointed out that although cycle to cycle monotonic progress with respect to the value of $d^t \bar{y}^*$ is achieved by this method, the finiteness argument given in this paper does not depend on that progress. Additionally, progress of this algorithm should make it feasible to drop particular hyperplanes from the definition of T_1 and S_1 after these hyperplanes become redundant to the progress of the algorithm. While detailed rules for dropping redundant hyperplanes have not been included here, there would appear to be no intrinsic conceptual problem barring development of such rules.

In cases where the function $d^t x = c^t x$, i.e., $d^t x$ is the objective function for the underlying integer program, it may be possible to considerably expedite termination of the algorithm. Suppose some feasible integer solutions in S are known and x^b is the best of these in terms of objective function value. Then by setting $\underline{w} = d^t x^b + 1$ we restrict the search of the algorithm to points in S that dominate x^b in terms of objective function value.

6. A Composite Method: Algorithm III.

Second level algorithms can be designed that use algorithms I and/or II in the rôle(s) of sub-routine(s). Algorithm III provides a specific example.

Algorithm III

1. Select vectors d^1, d^2, \dots, d^m and scalars u_1, u_2, \dots, u_m , so that the set, $\omega = \{y \mid (d^k)^t y \leq u_k\}$, satisfies $(S \cap \omega) \subset \theta$ and $S \cap \omega \cap \bar{\theta} = \emptyset$.
2. Set $i = 1$.
3. Commence execution of algorithm II, with $d = d^i$.
4. Continue iterations of algorithm II until an integer solution is located, or until $(d^i)^t \tilde{y}^* \leq u_i$ in some cycle of the θ -tangent phase. When the latter event occurs, go to 5.
5. If $i < m$, set $i = i + 1$ and go to 3; otherwise stop: S contains no integer solutions.

The termination in step 5 is justified because setting $T_m = \omega$ immediately yields a solution for algorithm I. Algorithm II is employed in steps 3 and 4 to achieve the collection of sufficiently deep cuts required in the definition of ω . There exist numerous ways to generate ω with the requisite properties. For efficiency it is important to choose the d^i, u_i $i = 1, \dots, m$ so that the total computational effort required in steps 2 and 3 is minimized. Thus the specification of the cuts that define ω is a substantial subproblem in which interesting possibilities exist for adaptation to problem classes, and to individual problems.

More flexible variants of this method can be imagined in which ω is specified dynamically - in the course of solving the problem, and in response

to specific structural characteristics of S and T - instead of the static, a priori specification in algorithm III.

7. Finiteness.

Both (primary) algorithms described in this paper can be shown to be finite on the basis of an elementary argument.

The foundation of this argument is the finite number of non-integral extreme points in S and the fact that each non-integral extreme point in S is a finite distance from $\bar{\theta}$.

Let x^k be an arbitrary non-integer extreme point of S . Let K be an index set for all non-integer extreme points of S .

Define:

$$H(x^k) \equiv \{h \mid h \in \bar{\theta} \text{ and } h^t \cdot x^k \geq h^t x \text{ for all } x \in S\},$$

$$\bar{v}(x^k) \equiv \max_{h \in H(x^k)} h^t x^k,$$

$$\bar{V} \equiv \max_{k \in K} \bar{v}(x^k).$$

Hence for all cuts developed in the S-cut phase we have $v_i < \bar{v}_i \leq \bar{V}$.

Note that for $k \in K$, $h^t x^k \geq p$ is impossible since $h^t x^k > p \Rightarrow x^k \notin \theta$ and $\theta \supset S$, while $[h^t x^k = p \text{ and } x^k \in \theta] \Rightarrow x^k \in \bar{\theta} \Rightarrow x^k$ is integral.

This shows that $\bar{V} < p$. For all h^i developed in the course of either algorithm we must have

$$(h^i)^t y^{i+m} \leq v_i \leq \bar{V} < p \quad \text{for all } m \geq 1.$$

Since $y^{i+m} = h^{i+m}$ we have

$$(h^i)^t h^{i+m} \leq \bar{V} < p.$$

Now suppose that either algorithm generates an infinite sequence of cycles. Then we have $\{h^i\}_{i=1}^{\infty}$. Since each $h^i \in \bar{\theta}$, and $\bar{\theta}$ is compact, $\{h^i\}$ must have a limit point. This means that $\|h^i - h^{i+m}\| < \epsilon$ for an arbitrary $\epsilon > 0$ and i sufficiently large. But

$$\|h^i - h^{i+m}\| = |(h^i)^t h^i + (h^{i+m})^t h^{i+m} - 2(h^i)^t h^{i+m}|^{\frac{1}{2}}$$

$$\|h^i - h^{i+m}\| = |2p - 2(h^i)^t h^{i+m}|^{\frac{1}{2}}$$

$$= |2(p - (h^i)^t h^{i+m})|^{\frac{1}{2}}$$

$$\geq |2(p - \bar{v})|^{\frac{1}{2}}$$

Since \bar{v} is a fixed constant less than p it is clear that $\|h^i - h^{i+m}\|$ has a fixed positive lower bound for all i . This contradicts our assumption of an infinite sequence of cycles.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

8. Related Work.

The papers on convexity cuts, intersection cuts, and hypercylindrical cuts [1], [2], [5] constitute a general background and source of concepts for this paper. More specifically the Glover and Klingman paper [3], which develops a finite convexity cut method (based on Tui's work [4]), provides implicitly an instance of the basic conceptual strategy employed here. The common concept employed in both papers is the development of a set that eventually is between S and $\bar{\theta}$. In our paper T_i plays this role. We maintain the relation $T_i \supset S_i$ in every cycle and make progress with each cycle toward the goal of $T_i \cap \bar{\theta} = \emptyset$. The "in between" set in the Tui, Glover and Klingman development, denoted by D_i , is always in θ ; and while D_i may intersect $\bar{\theta}$, there are no integer points in $D_i \cap S$ (since if such exist they are automatically discovered and the (sub) problem is solved). In their method progress is made toward the goal $D_i \supset S$; and the progressive alteration of D_i is one of expanding the set D_i in each stage when it is discovered that a hyperplane boundary of D_i cuts S . Thus this paper and the Tui, Glover and Klingman papers illustrate particular tactical plans for realizing the general strategic goal of constructing a set that is appropriately between S_i and $\bar{\theta}$. It is clear that other combinatorial variants of these tactics are possible within the framework of the same strategic goal.

9. Loose Ends and Speculation.

As has already been indicated at various points in this paper there are several opportunities for generating new algorithms by varying certain elements of the given algorithms or by the construction of composite algorithms. Some further opportunities of this sort will be listed here.

- It may prove useful to employ other convex sets in the role played by θ in this paper -- see [2] and [1] .
- The cut employed here is parallel to h^1 , the objective function (normal) used as a "target" in locating x^1 . This option was selected mostly for expositional convenience. The usual hyperspherical cut (available from x^1), which is typically a deeper cut, can also be employed.
- It would appear possible to develop variations of algorithm I in which partially randomized search is used in the θ -tangent phase.

No computation has been attempted with any of the three algorithms.

Speculating, it would appear that algorithm I might be more efficient at locating integer solutions -- particularly if they are relatively plentiful -- than in demonstrating that none exist in S . Thus it appears to have better prospects in the "primal" role of locating nearly optimal or optimal solutions expeditiously. Algorithm II, alternatively, may be more useful in determining that an optimum has been obtained when it is employed with $d = c$ and with \underline{w} set equal to the next integer above the objective function value of the best known integer solution in S .

Since the efficiency of these methods would appear to depend essentially on geometric relations involving S and θ , it is possible that they may provide additional motivation and means to more effective classification of problems in terms of characteristics that influence ease of solution.

10. Acknowledgements.

I want to express my thanks to Dr. R. M. Thrall for his helpful discussion of several topics related to this paper.

This research was supported by NASA Grant *NGR-44-006-128*.

REFERENCES

- [1] Balas, Egon, "The Intersection Cut - A New Cutting Plane for Integer Programming," Operations Research, 19:1, Jan.-Feb., 1971, pp. 19-39.
- [2] Glover, Fred, "Convexity Cuts," working paper, School of Business, University of Texas, Austin, Texas, December 1969.
- [3] Glover, F. and D. Klingman, "Concave Programming Applied to a Special Class of Zero-one Integer Programs," Series in Applied Mathematics for Management. Publication No. AMM-11, Graduate School of Business, University of Texas, Austin, January 1969.
- [4] Tui, Hoang, "Concave Programming Under Linear Constraints," (Russian) Doklady Akademii Nauk SSSR, 1964. English translation in Soviet Mathematics, 1964, pp. 1437-1440.
- [5] Young, R. D., "Hypercylindrically Deduced Cuts in Zero-One Integer Programs," to appear in Operations Research.